금융데이터분석

<Text Mining 기법 적용을 통한 포트폴리오 설계 및 시각화>

팀명
4조

팀원

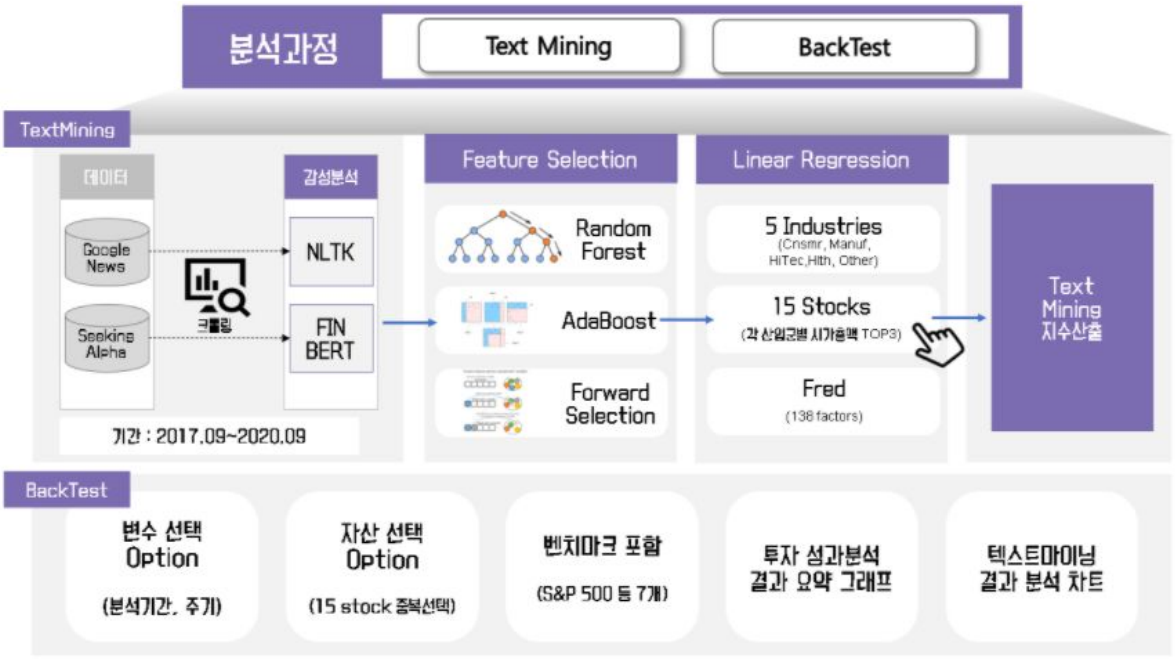
2014100909 산업경영공학과 이근영
2017103727 소프트웨어융합학과 박채영
2015100901 산업경영공학과 김경덕

1. 프로젝트 소개 및 목적

프로젝트의 목적은 크게 세가지입니다. 첫째, 금융데이터 수업의 기법들을 복습하고 코드로 구현해보며, 둘째, 장고를 통해 시각화와 사용자가 커스터마이징을 편리하게 설계하여 자유도를 높이는 것입니다. 또한 확장성을 높일 수 있습니다. 셋째, 텍스트 마이닝과 Feature selection을 활용하여 여러가지 정보를 도출하고 이를 포트폴리오 설계에 활용하는 것입니다.

2. 프로젝트 개요(분석 방법론)

프로젝트의 전체 개요는 아래 사진과 같습니다. 프로젝트는 크게 Text Mining 분석과 BackTest로 분리할 수 있습니다.



먼저 Text Mining 분석을 살펴보겠습니다. Text Mining을 위한 데이터를 크롤링하여 수집하였습니다. 데이터는 Google News와 Seeking Alpha site에서 크롤링하였습니다. 그 후 감성 분석을 진행하였는데, Google News의 경우 NLTK를 Seeking Alpha의 경우 FIN BERT를 사용하였습니다. 감성 분석 결과 산출된 파라미터가 유의미한지 살펴보기 위해 회귀분석을 사용하였는데, 이때 사용되는 변수가 너무 많아서 feature selection 과정을 거쳤습니다. feature selection은 random forest, Adaboost, Forward Selection 을 사용하였습니다. Linear Regression 결과 감성 분석의 파라미터가 return값 예측에 유의미한 영향을 끼친다는 사실을 발견하였고, 이를 활용하여 text mining 지수를 산출하였습니다. text mining 지수는 추후 Portfolio의 constraints에 반영되어 Text Mining Portfolio를 만드는데 사용하였습니다.

두번째는 BackTest입니다. BackTest 구현은 Django를 이용하여 웹구현을 하였습니다. 사용자는 분석기간과 주기, 자산의 종류를 직접 선택할 수 있습니다. 사용자의 선택 이후에는 투자 성과분석 결과 요약 그래프 등을 확인할 수 있습니다.

3. TEXT MINING 및 회귀분석

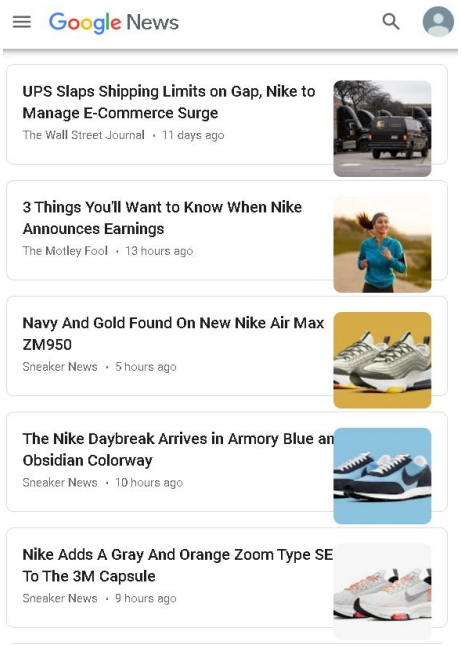
이번 장에서는 텍스트 마이닝을 위한 3.1 데이터 수집과 전처리, 그리고 이를 이용한 3.2 감성분석과 회귀분석을 위한 여러 독립변수들의 3.3 Feature selection 그리고 선택된 변수를 이용한 3.4 회귀분석 으로 이루어져 있으며 앞선 3.2와 3.4의 결과를 바탕으로 3.5 텍스트 마이닝 지수 산출로 이루어져 있습니다.

3.1 데이터 소개 및 수집 방법

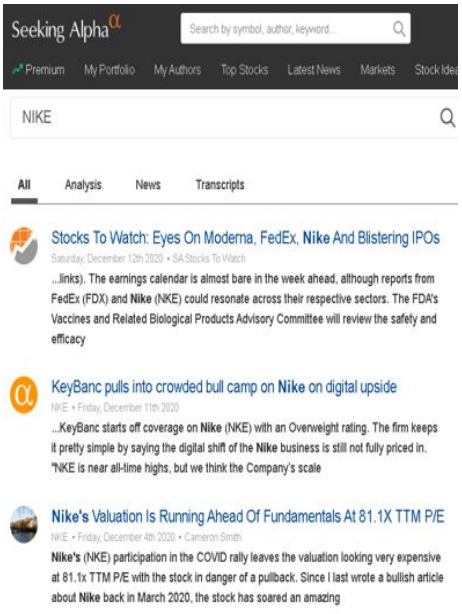
데이터의 경우 크게 3가지의 데이터가 목적별로 수집하였습니다. 첫째, 뉴스 기사 크롤링 데이터 둘째, 회귀분석의 독립변수로 사용 할 거시경제 데이터 및 Fama French 데이터 셋째, 종속변수로 사용 할 여러 수익률 데이터 입니다.

(1) 뉴스 기사 크롤링

뉴스 기사 데이터는 일반 기사 데이터가 포함된 구글 뉴스와 대부분이 주가와 관련이 높은 금융 경제기사를 제공하는 사이트인 Seeking Alpha로 부터 크롤링을 하였습니다. 좌측의 사진과 같이 NIKE검색어를 구글뉴스창에 입력하면 나오는 결과를 크롤링하는 코드를 구현하였으며, 이때 설정할 수 있는 항목은 시작과 끝의 기간, 키워드가 있으며 코드 실행시 구글의 자동화 로봇 감지 프로그램이 작동하여 이를 해결하기 위해 한달분량의 뉴스기사를 크롤링 완료하면 로그아웃을 한 후 자동로그인을 하도록 하는 코드를 구현하여 이러한 문제를 해결 하였습니다.



다음은 Seeking Alpha의 news 크롤링 입니다. 구글의 하단 버튼을 눌러 다음 페이지의 뉴스기사를 보는 방식과 달리 스크롤링 하는 방식으로 설정한 기간 전체의 기사를 볼 수 있어 구글 뉴스에 비해 매우 빠르게 수집이 가능하였으며, 이는 크롤링 코드에 스크롤을 정해진 초마다 실시하며 일정 기간동안 스크롤에도 새로운 기사가 나오지 않으면 멈추고 지금까지 스크롤한 모든 기사를 긁어오는 방식으로 진행하였습니다. 구글과 Seeking Alpha모두 2017.09~2020.09의 데이터를 수집하였습니다.



(2) 회귀분석 독립변수 데이터

다양한 수익률을 종속변수로 한 회귀분석을 진행하기 위해 텍스트 마이닝의 감성분석 변수외에도 비교를 위한 다양한 변수를 추가하였으며 이는 금 선물 시장을 뉴스 감성지수와 거시경제 데이터를 활용하여 회귀분석을 진행한 논문을 참고하였습니다. 분석 프레임 외에 사용된 거시경제 데이터의 경우 본 연구에서는 선행논문과 달리 개별종목에 대해 연구를 진행하였기 때문에 휴리스틱한 과정을 통해 수집하기 보다는 최대한 많은 거시경제 데이터 속에서 개별종목과 관련성이 큰 거시경제 데이터를 Feature Selection하여 사용하였습니다. 이때 사용된 거시경제 데이터는 [1]의 논문을 참조하여 138개의 데이터를 수집하였으며 좌측의 사진은 거시경제 데이터 중 Money and Credit section의 데이터 입니다. 이러한 변수중에서 실제 회귀분석에 활용한 변수는 Random Forest, AdaBoost, Forward Selection의 과정을 통해 각 종목별 관련 변수를 선정하였습니다. 수집한 데이터는 1959년 부터 현재 까지의 월별데이터 이며, 텍스트 마이닝 변수와 함께 사용하기 위해 2017에서 2020년 까지의 데이터만 사용하였습니다.

fred	description
M1SL	M1 Money Stock
M2SL	M2 Money Stock
M2REAL	Real M2 Money Stock
BOGMBASE	Monetary Base
TOTRESNS	Total Reserves of Depository Institutions
NONBORRES	Reserves Of Depository Institutions
BUSLOANS	Commercial and Industrial Loans
REALLN	Real Estate Loans at All Commercial Banks
NONREVSL	Total Nonrevolving Credit
CONSPI	Nonrevolving consumer credit to Personal Income
MZMSL	MZM Money Stock
DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding
DTCTHFNM	Total Consumer Loans and Leases Outstanding
INVEST	Securities in Bank Credit at All Commercial Banks

거시경제 데이터 외에도 팩터를 통해 개별 주식을 설명할때 많이 사용 되는 [2] Fama French의 MKT, SMB, HML 팩터를 [3]위 홈페이지와 python의 fama french library로 수집하여 사용하였습니다.

(3) 회귀분석 종속변수 데이터

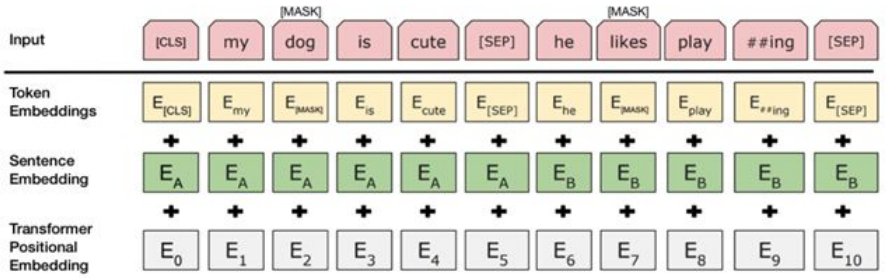
회귀분석은 크게 세가지 종속변수 카테고리로 구성하였으며, 각각은 5개의 산업군 수익률 데이터, 텍스트 마이닝을 실행한 15가지의 종목의 수익률 데이터, 15가지 수익률 데이터를 산업군에 맞게 분류한 후 이들의 수익률을 평균한 5개 Sector 수익률 데이터입니다. 이때 5개의 산업군 수익률 데이터는 fama french library 5 industry 데이터로부터, 각 종목의 수익률은 Pandas DataReader의 Yahoo library로 부터 수집하였으며, 벤치마크로 사용할 S&P500 수익률 데이터 또한 Yahoo로 부터 수집하였습니다.

3.2 감성 분석 기법

[1] McCracken, Michael W., and Serena Ng. "FRED-MD: A monthly database for macroeconomic research." *Journal of Business & Economic Statistics* 34.4 (2016): 574-589.
[2] Bahl, Bhavna. "Testing the Fama and French three-factor model and its variants for the Indian stock returns." *Available at SSRN 950899* (2006).
[3] https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

본 연구에서는 구글 뉴스 기사와 Seeking Alpha의 기사의 크롤링을 진행하였습니다. 또한 이렇게 수집된 크롤링 데이터를 finBERT 기법을 활용하여 기사의 긍정과 부정의 정도를 판정하고 이를 여러 변수로 재가공 하는것을 목표로 하였습니다.

(1) BERT 모형



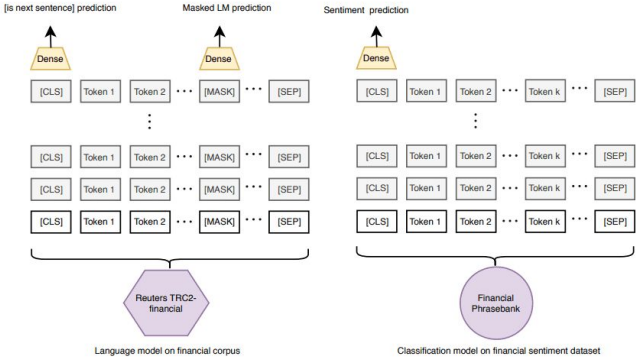
BERT모델[4]은 대용량 코퍼스 데이터를 활용하여 언어 전반에 대해 학습을 합니다. 학습에 사용되는 인풋데이터는 각각 word token embedding, segment embedding, position embedding로 파생되어 대량의 데이터 규모로 사전 학습됩니다. 이렇게 1차적으로 기 학습된 모델을 활용하여 사용할 목적에 맞게 fine-tuning(2차 학습)하여 해당 도메인에 더욱 좋은 성능을 낼 수 있습니다. 앞선 대량의 코퍼스 데이터를 학습하는 단계에서 word token embedding의 경우 sentencepiece로 토큰화되고, BERT 논문에 삽입된 그림 1와 같이 문장 시작부분에는 [CLS], 개별 단어간의 분리는 ##기호와, 문장 종료시에는 [SEP] 토큰을 붙여 입력 데이터를 구성하게됩니다.

(2) finBERT 모형

본 연구에서는 기본적인 BERT모형 대신, 금융도메인에 최적화된 모델인 finBERT모델[5]을 활용하여 연구를 진행하였습니다.

finBERT모델은 financial text의 경우 unique vocabulary를 가진 특수한 Terminology가 있고 쉽게 식별되는 긍/부정 단어 대신 비유와 같은 모호한 표현을 사용하는 경향이 있기 때문에 일반적인 코퍼스 데이터로 학습된 모델을 금융 도메인에 사용하기에는 적합하지 않다는 문제를 해결하기 위해 고안된 모델입니다.

따라서 앞서 소개한 BERT모델에서 1차 학습 대용량 코퍼스 데이터의 경우 Reuters에서 발행한 1.8M 뉴스 기사로 구성된 Reuters TRC2의 subset을 이용하여 1차적으로



[4] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
[5] Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." *arXiv preprint arXiv:1908.10063* (2019).

finance 도메인에 맞는 언어능력을 학습을 진행하였습니다. 이어서 2차 학습의 경우 LexisNexis database의 financial news에서 무작위로 선택된 4845개의 영어 문장에 긍부정 라벨이 붙은 데이터를 활용하여 진행하였습니다.

이때, 논문에서 제시한 finBERT 가상환경을 구축하고 여기에 BertTokenizer, BertForSequenceClassification, PYTORCH_PRETRAINED_BERT_CACHE 라이브러리를 사용하여 분석을 진행했습니다.

(3) 감성판정

크롤링을 통해 수집한 구글 뉴스와 Seeking Alpha의 뉴스 데이터는 finBERT모델을 통해 감성변수를 생성하였으며 감성분석을 진행한 직후의 Raw Data의 예시는 좌측의 사진과 같습니다. 여기서 각 레코드는 기사 하나를 의미하는데 이에 대해 sentiment 지표인 가장 4열의 데이터를 활용하여 3열의 positive, negative, neutral의 감성판정을 진행하며, 이들 변수는 sentiment 지표의 경우 월별 평균, 감성판정결과의 경우 각각 월별 개수를 집계하여 변수로 생성하였습니다.

Tim Cook scores first major st	[0.540645	positive	0.479456
Apple accused of delaying bl	[0.026180	negative	-0.82072
BofA names the defensive stc	[0.040581	neutral	0.02235
Apple's largest assemblers inv	[0.067814	neutral	0.052654
Apple pauses 30% cut on real	[0.036964	negative	-0.84256
Buy the dip in Apple ahead o	[0.171812	neutral	0.040751
EU appeals Apple tax ruling tc	[0.043724	negative	-0.61363
"Spotify, Match Group, and E	[0.316777	neutral	0.303366
Apple sidelined at UBS on 'un	[0.013842	negative	-0.93216
Apple launches first online stc	[0.548657	positive	0.503515

Company	Date	mean_g	pos_g	neg_g	neu_g	mean_s	pos_s	neg_s	neu_s	neg_g_r	neg_s_r
GOOGL	2017-09-01	0.036231	15.0	8.0	73.0	-0.112671	3	12	38	0.347826	0.800000
GOOGL	2017-10-01	0.075615	21.0	10.0	61.0	-0.043619	5	7	26	0.322581	0.583333
GOOGL	2017-11-01	0.005836	17.0	16.0	62.0	-0.375496	0	11	16	0.484848	1.000000
GOOGL	2017-12-01	0.041672	23.0	11.0	62.0	0.030489	5	6	13	0.323529	0.545455
GOOGL	2018-01-01	0.106406	26.0	8.0	62.0	0.000831	7	7	23	0.235294	0.500000
...	...	...	...	...	...	...	...	...	...	...	...
WMT	2020-05-01	-0.017700	17.0	22.0	57.0	-0.152835	4	12	5	0.564103	0.750000
WMT	2020-06-01	-0.039328	17.0	23.0	54.0	-0.107880	4	5	8	0.575000	0.555556
WMT	2020-07-01	-0.023563	14.0	19.0	57.0	-0.058508	5	6	10	0.575758	0.545455
WMT	2020-08-01	0.030407	20.0	18.0	56.0	-0.096650	4	9	15	0.473684	0.692308
WMT	2020-09-01	0.023461	23.0	16.0	50.0	0.101943	11	7	28	0.410256	0.388889

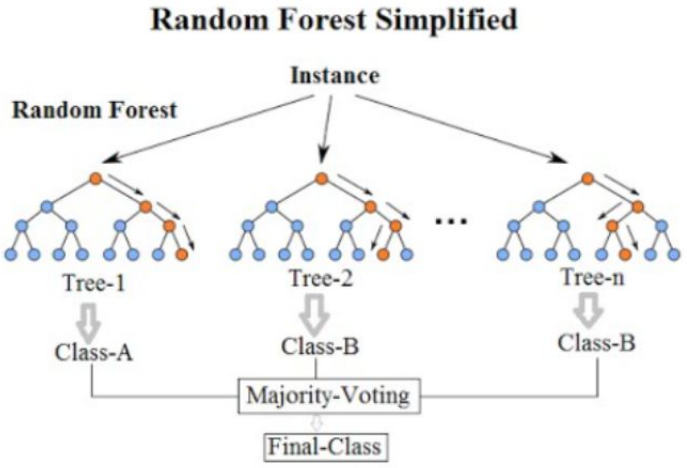
위 테이블은 최종적으로 분석의 사용된 텍스트 마이닝 변수들입니다. 변수명 뒤의 _g, _s는 각각 google과 seeking alpha를 의미합니다. mean변수의 경우 앞서 언급한 sentiment 지표의 월별 평균을 의미하며, pos, neg, neu는 각 감성판정 결과의 월별 개수의 집계 개수를 의미합니다. 여기서 추가적으로 부정 뉴스기사의 비중을 파악하기 위해 전체 기사 개수에서 부정기사가 차지하는 정도를 _r의 변수명이 붙은 파생변수를 생성하였습니다.

3.3 Feature selection 기법

회귀분석을 진행하기에 앞서 변수가 너무 많은 문제를 해결해야 했습니다. 예를 들어 회귀분석에 사용된 독립변수 중 하나인 Fred 데이터의 경우 138개의 변수를 가지고 있었습니다. 이를 분석에 모두 사용하게 되면 과적합이 발생하기 때문에 이를

방지하기 위해 feature selection을 진행하였습니다. Feature selection은 3가지 방법을 사용하였는데 각각 Random Forest, Adaboost, Forward Selection 입니다.

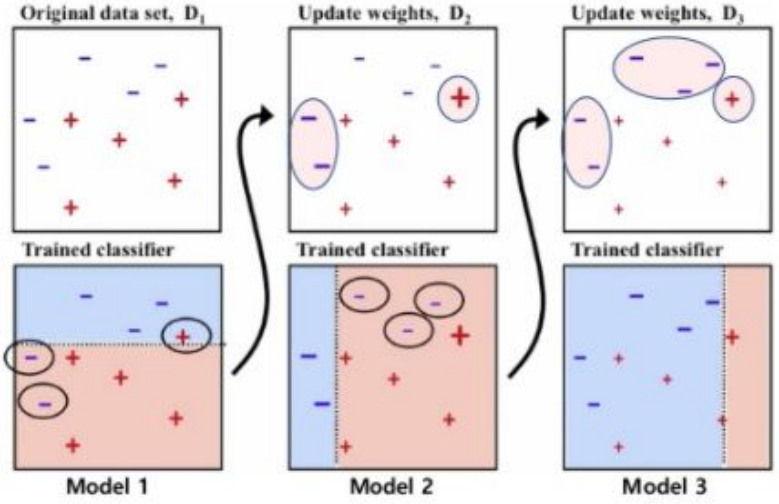
(1) Random forest



Random forest는 간단하게 말하자면 다수의 결정트리를 학습하는 앙상블 방법입니다. feature space를 랜덤하게 쪼개고, 쪼개진 sub feature space를 다시 decision tree를 이용해 격자 형태의 decision boundary를 갖도록 분류합니다. 이렇게 학습한 N 개의 decision tree를 앙상블하여 최종 추론을 합니다. 즉, 추론할 때에는 데이터를 모든 decision tree에 집어넣고 각각의 분류결과를 다수결(majority voting)의 방식으로 최종 결과를 합니다.

Random Forest의 feature importance를 기준으로 feature selection을 진행하였고, sklearn library를 이용하여 분석을 진행하였습니다. 여기서 feature importance란 각각의 attribute가 얼마나 분류 성능에 영향을 미치는지, 즉 feature의 유용성을 판단하는 지표입니다.

(2) Adaboost



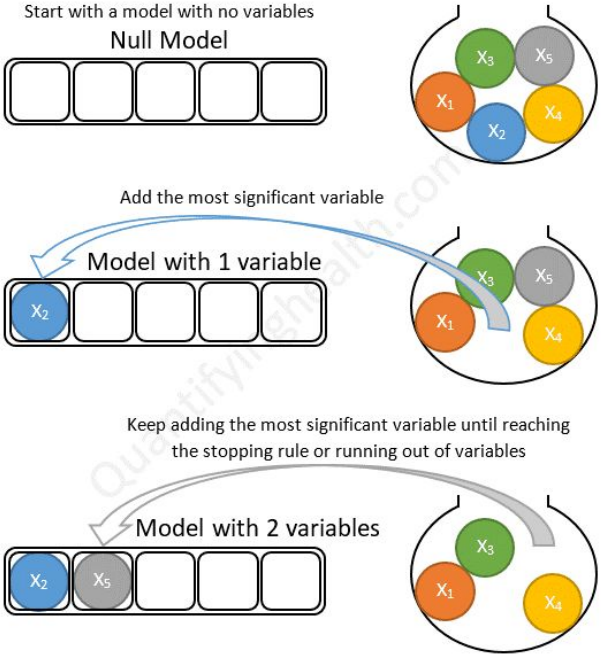
Adaboost는 다수결을 통해 정답을 분류하고 도출되는 오답에 가중치를 조절하는 방법입니다. 위의 그림을 통해 간단히 설명하자면, model 1에서 데이터를 예측하고, 잘못 예측한 데이터에 가중치를 부여합니다. model2는 앞선 모델에서 잘못 예측한 데이터에 가중치가 부여되었으므로 자동으로 잘못

예측한 데이터를 올바르게 분류하는데 집중하게 됩니다. 마찬가지로 model2의 결과를 보고 잘못 예측한 데이터에 가중치를 부여하는데, model3는 앞선 model1,2에서 잘못 예측한 데이터를 분류하는데 집중하게 되는 원리를 가지고 있습니다. 최종적으로 모델별로 계산된 각각의 가중치를 합산하여 최종 모델을 생성하는 방법입니다.

Random Forest와 마찬가지로 model의 feature importance를 기준으로 feature selection을 진행하였고, sklearn library를 이용하여 분석을 진행하였습니다.

(3) Forward Selection

Forward stepwise selection example with 5 variables:

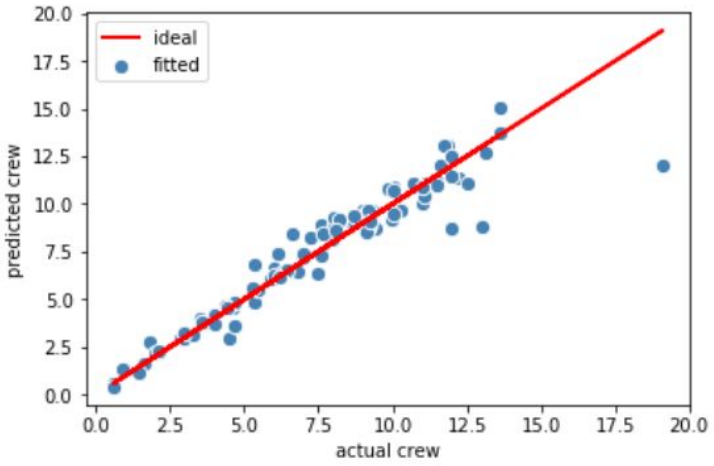


Forward selection은 모델에 파라미터를 하나씩 넣어가면서 적합도가 가장 좋은 모델을 찾는 방법입니다. 하나씩 파라미터를 추가할 때 가장 중요한 변수를 선택하는 방법을 택하고, 고려할 나머지 모든 변수가 model에 추가되었을 때 일부 임계값보다 큰 p-value를 가질때 멈추게 됩니다.

3.4 Linear Regression

3.3 의 feature selection을 통해 선택된 변수들과 텍스트마이닝 변수를 독립변수로, 각 종목과 5개 산업군을 종속변수로 두고 선형 회귀분석을 진행하였습니다.

(1) Linear Regression



Linear Regression이란 위의 이미지와 같이 어떤 dataset내에서의 어떤 특성을 Linear로 표현하는 과정을 뜻합니다.

(2) Linear Regression 사용 이유

텍스트마이닝 지수 산출을 위해 linear regression을 사용한 이유는, 금 선물 시장과 뉴스 척도의 관계를 살펴보기 위해 텍스트 마이닝 변수와 거시경제 변수를 함께넣고 선형 회귀분석을 진행한 연구[6]와 S&P500을 뉴스 인덱스와 P/E ratio를 사용해 회귀분석한 연구[7]를 참조하였습니다. 본 연구에서 linear regression의 통계 지표들을 살펴보는 것 또한 의미가 있지만 이러한 결과 중 몇가지 사례를 선정하여 감성분석 결과 산출된 파라미터가 유의미한지 살펴보기 위한 지표로 사용하였습니다.

(3) 회귀분석 방법 설명(5 Industries, 15 stocks, fred)

회귀분석의 종속변수를 설정하는 방법은 크게 세가지로 나눌 수 있습니다. each stock과 5 industry, 그리고 5 sector의 수익률을 종속변수로 사용하였습니다. each stock의 경우 15개의 stock(Amazon, Walmart, Nvidia, Tesla, P&G, Nike, Apple, MicroSoft, Facebook, Johnson & Johnson, Pfizer, Merck & Co,Alphabet, Berkshire hathaway, Visa)을 모두 사용하였습니다. 5 industry의 경우 fama french에서 가져와서 사용하였고, 5 sector의 경우 15개의 stock에서 5개 산업군별로 3개씩 평균을 계산하여 사용하였습니다. 정리하자면, 5 industry와 5 sector의 차이는 5 industry의 경우 무수히 많은 stock들이 포함되어 있지만, 5 sector의 경우 시가총액 기준 top3의 stock만 포함되어 있다는 점입니다.

회귀분석의 독립변수를 설정하는 방식 또한 세가지로 나눌 수 있습니다. Fama french 3 factor, Random Forest, Adaboost입니다. Fama french 3 factor의 경우 MKT, SMB, HML 의 3가지 factor를 사용하였고, Random Forest와 Adaboost의 경우 앞서 진행했던 feature selection을 통해 추출된 변수를 사용하였습니다.

---

[6] Smales, Lee A. "News sentiment in the gold futures market." *Journal of Banking & Finance* 49 (2014): 275-286.

[7] Arvanitis, Konstantinos, and Nick Bassiliades. "Real-time investors' sentiment analysis from newspaper articles." *Advances in combining intelligent methods*. Springer, Cham, 2017. 1-23.

지금까지 종속변수와 독립변수의 경우의수가 각각 3가지씩 존재하므로, 총 진행한 회귀분석의 조합수는 3 X 3 = 9 조합이 발생합니다. 본 프로젝트에서는 9개의 조합을 3가지 방법으로 변형하여 총 3 X 3 X 3 = 27조합으로 회귀분석을 진행하였습니다.

변형된 3가지 방법은 각각 기본모형, Text Mining 변수 추가, Text Mining 변수 추가 + forward Selection입니다. 기본 모형의 경우 설정된 종속변수와 독립변수를 그대로 사용한 모델입니다. Text Mining 변수 추가의 경우 감성분석을 통해 산출된 변수를 추가하였습니다. 추가된 변수의 예시로는 neg_g(google news의 negative 빈도), neu_g,(google news의 neutral 빈도) pos_g(google news의 positive 빈도) 등이 있습니다. Text Mining + forward selection의 경우 감성분석을 통해 산출된 변수를 추가한 후, 기본 base가 되는 모델은 regression으로 하면서 forward selection을 통해 변수를 하나씩 넣어가면서 가장 설명력이 좋은 조합의 모델을 찾는 방식으로 분석을 진행하였습니다.

(4) 회귀분석 결과 - 채영

27가지 조합의 회귀분석을 진행하였는데, 그 결과는 아래 이미지와 같습니다. 아래 이미지의 여러 조합의 독립변수를 사용한 회귀분석 진행의 각 셀의 값이 의미하는 바는 열이름인 기본적인 독립변수와 행이름인 종속변수, 그리고 이를 각각 기본 독립변수외에 아무것도 추가하지 않은 첫째 기본모형과, 이에 텍스트 마이닝 변수를 추가한 둘째 Text mining 변수추가, 기본모형과 Text mining변수를 추가한 상태에서 forward selection을 진행한 셋째 모형의 결과이다. 이때, 각 셀의 값은 각 모형을 취했을때 나오는 15개의 종목의 회귀분석의 adj R^2값의 평균을 의미한다.

기본모형	Fama French Three	Random Forest	Ada Boost
Each stock	46.13%	29.48%	27.29%
Industry 5	90.52%	40.42%	43.60%
5 sector	61.59%	33.86%	38.40%

Text mining 변수추가	Fama French Three	Random Forest	Ada Boost
Each stock	45.54%	24.94%	26.75%
Industry 5	89.70%	40.42%	31.07%
5 sector	64.50%	31.04%	34.71%

TM + Forward Sel	Fama French Three	Random Forest	Ada Boost
Each stock	51.99%	36.78%	35.00%
Industry 5	91.19%	51.13%	52.56%
5 sector	58.73%	46.61%	48.45%

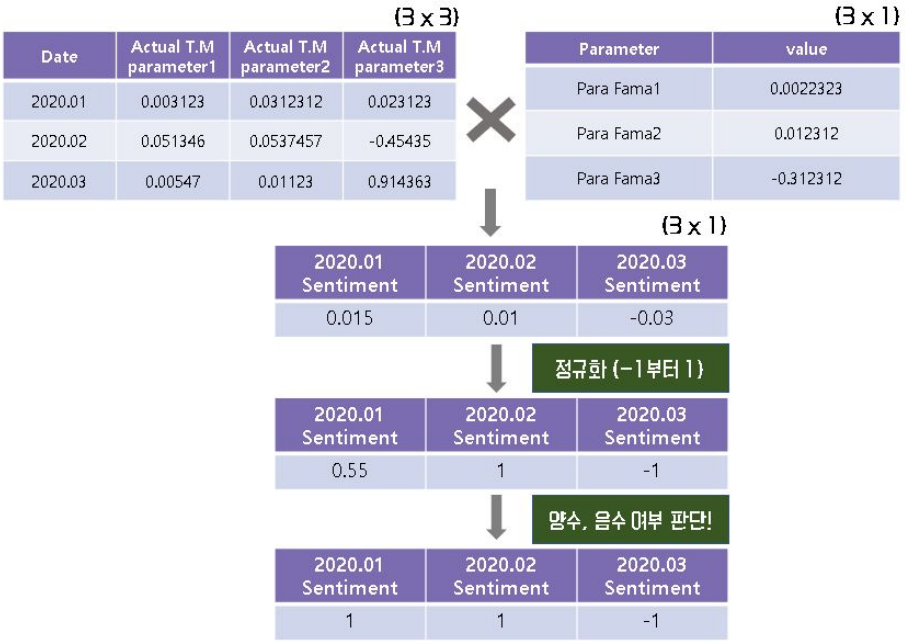
여러 조합의 독립변수를 사용한 회귀분석 진행

	AMZN	WMT	NVDA	TSLA	PG	NKE	AAPL	MSFT	FB	JNJ	PFE	MRK	BRK-A	GOOGL	V
Adj R^2	62.96%	35.34%	55.93%	40.20%	32.29%	57.60%	55.04%	67.38%	56.09%	51.91%	26.80%	18.01%	73.38%	76.28%	70.70%
p-value	0.01	0.02	0.03	0.01	0.03	0.01	0.004	0.03	X	X	0.03	0.06	0.02	0.0003	0.06
params	0.002	0.007	-0.0071	0.713	0.0188	0.233	0.278	-0.233	X	X	0.00646	-0.289	-0.00454	0.620	0.00272

선정된 회귀모형의 결과

(5) 감성 지표 생성

3.3의 (3)회귀분석 방법 설명(5 Industries, 15 stocks, fred)에서 여러가지 회귀분석을 진행하였고, (4)회귀분석 결과에 이를 요약하였습니다. 우선 (4)의 세가지 모형 중 기본모형에 Text Mining 변수 추가하고 forward Selection 을 진행한 케이스의 평균적인 adj R^2값이 우수하여 이를 채택하여 추가적인 분석을 진행하였으며, 이때 Each stock에 대해 Fama French 변수를 함께 회귀분석한 결과의 R^2 개선도가 가장 높아 이 케이스 하나에 대해서만 추가적인 감성 지표를 생성 하였습니다.
감성지표의 경우 아래 이미지의 프로세스를 따라서 산출하였습니다. 아래의



프로세스는 선정된 회귀분석을 통해 나온 계수를 좌측의 (3x1) 메트릭스의 예시처럼 받은 후 실제 텍스트 마이닝의 값인 예시의 (3x3) 메트릭스와 같이 (nx3)형태의 메트릭스와 곱하여 n개의 일자별 감성지표를 생성하며 이러한 작업을 15개의 종목에 대하여 반복적으로 진행합니다. 이러한 감성지표 생성이 끝난후에는 지표에 대해 전처리를 해주었는데, 이는 -1에서 1사이의 정규화를 하였고 이후 실제 포트폴리오에 적용시에는 양수와 음수로 구분하여 사용하였습니다.

(6) 감성 지표 활용

3.3의 (5) 감성 지표 생성의 단계를 통해 도출된 감성지표의 실제 값의 예시는 아래의 이미지과 같으며 이러한 감성지표값은 각 종목에 대해 포트폴리오의 가중치를 조절하는데에 사용되었습니다. 이러한 감성 분석 지수를 직접

Date	AMZN	WMT	NVDA	TSLA	PG	NKE	AAPL	JNJ	GOOGL	MSFT	PFE	BRK-A	FB	MRK	V
2017-09-01	-0.02123	-0.04006	0.047757	0.180503	-0.03168	-0.0168	0.037674	-0.01347	0.0159	0.038489	0.058297	-0.02149	0.037959	0.022754	0.008214
2017-10-01	0.068983	0.04284	0.074345	0.129767	0.019019	0.06494	0.070392	0.028563	0.060502	0.043187	-0.0081	0.024393	0.034543	-0.0149	0.044159
2017-11-01	0.103845	0.056994	0.054359	-0.05762	0.002083	0.057663	0.048567	0.026174	0.034315	0.028495	0.017131	0.032536	0.045972	0.01629	0.044356
2017-12-01	0.037616	0.013204	0.018063	0.215689	0.007453	0.025711	-0.00321	0.014796	0.024897	0.040291	0.009774	0.026968	0.018911	-0.02503	0.01622
2018-01-01	0.185753	0.067978	0.195929	0.093068	0.038383	0.133138	0.052466	0.062941	0.111389	0.109863	0.027172	0.082176	0.078288	0.024284	0.079912
2018-02-01	-0.04809	-0.01747	-0.02308	0.003212	0.004943	0.030179	-0.0053	-0.03362	-0.04381	0.006906	-0.0283	-0.04701	-0.04296	0.003029	-0.01471
2018-03-01	-0.03907	-0.04982	-0.0009	-0.06704	-0.02717	-0.03362	0.014479	-0.04911	-0.0152	0.009182	0.012755	-0.05573	-0.02588	-0.00167	0.003883
2018-04-01	-6.61E-05	0.005461	-0.08044	-0.02045	-0.02038	0.021805	-0.02146	-0.00816	0.004791	0.011933	-0.00186	0.012	0.008796	0.057407	0.008462
2018-05-01	0.07572	-0.03236	0.082785	-0.01129	0.001602	0.047794	0.097773	-0.01793	0.032991	0.040658	0.013977	-0.02569	0.039798	0.039042	0.032258
2018-06-01	0.027272	-0.02491	0.071326	-0.12409	0.018681	0.022082	0.050742	-0.00693	0.013362	0.02907	0.012348	0.004436	0.011292	0.002176	0.014298
2018-07-01	0.06593	0.043736	0.012176	0.055402	0.035346	0.027071	0.064162	0.037872	0.060925	0.039478	0.049928	0.05337	0.046892	0.02358	0.031181

최적화 포트폴리오의 제약식에 추가한 경우는 없었지만 선행연구[8]에서 SNS의 언급빈도를 포트폴리오의 목적식에 가공하여 넣은 사례가 있다는 것을 감안하여 진행하였습니다. 실제 제약식에 추가시에는 각 종목별로

```
if (symbols[0] in stock):
    if sentiment_ox[1] > 0:
        min_sec1 = 0.05
        max_sec1 = 1
    elif sentiment_ox[1] < 0:
        min_sec1 = 0
        max_sec1 = 1 / len(stock)
    else:
        min_sec1 = 0
        max_sec1 = 1
else:
    min_sec1 = 0
    max_sec1 = 0
```

```
w = cp.Variable(asset_num)
objective = cp.Minimize(cp.quad_form(w, sigma))
constraints = [cp.sum(w) == 1, w >= 0,
w[0] >= min_sec1,max_sec1 >= w[0],
w[1] >= min_sec2,max_sec2 >= w[1],
w[2] >= min_sec3,max_sec3 >= w[2],
w[3] >= min_sec4,max_sec4 >= w[3],
w[4] >= min_sec5,max_sec5 >= w[4],
w[5] >= min_sec6,max_sec6 >= w[5],
w[6] >= min_sec7,max_sec7 >= w[6],
w[7] >= min_sec8,max_sec8 >= w[7],
w[8] >= min_sec9,max_sec9 >= w[8],
w[9] >= min_sec10,max_sec10 >= w[9],
w[10] >= min_sec11,max_sec11 >= w[10],
w[11] >= min_sec12,max_sec12 >= w[11],
w[12] >= min_sec13,max_sec13 >= w[12],
w[13] >= min_sec14,max_sec14 >= w[13],
w[14] >= min_sec15,max_sec15 >= w[14]]
problem = cp.Problem(objective, constraints)
problem.solve(solver=cp.ECOS)
```

Max와 Min가중치를 감성 지표에 따라 다르게 설정하였습니다. 이때 좌측의 if문을 보면 감성지수가 좋은경우 최소한 5%이상의 비중을 가져가도록 하였고, 감성지수가 좋지 않은 경우 Max 제약 비중을 1/stock의 개수로 가져 가도록 하였습니다. 이러한 과정에서 제약식의 비중 제한 조건의 근거는 선행연구가 없어 임의로 판단하여 선정하였습니다.

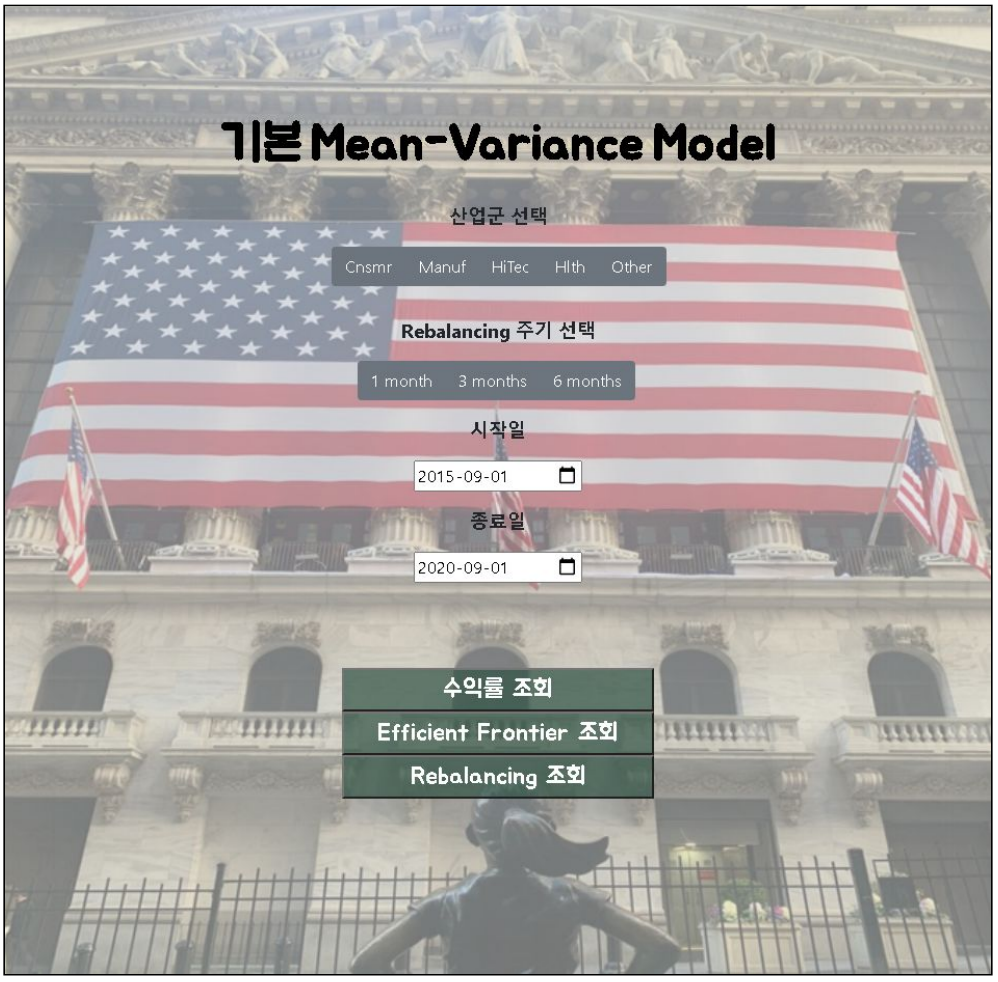
[8] Sun, Andrew, Michael Lachanski, and Frank J. Fabozzi. "Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction." *International Review of Financial Analysis* 48 (2016): 272-281.

4. BackTest

BackTest 시연은 Django를 이용하여 구현하였습니다. 기능을 크게 두가지로 나누자면, 기본 M-V 모델과 TextMining 모델 기능이 있습니다.

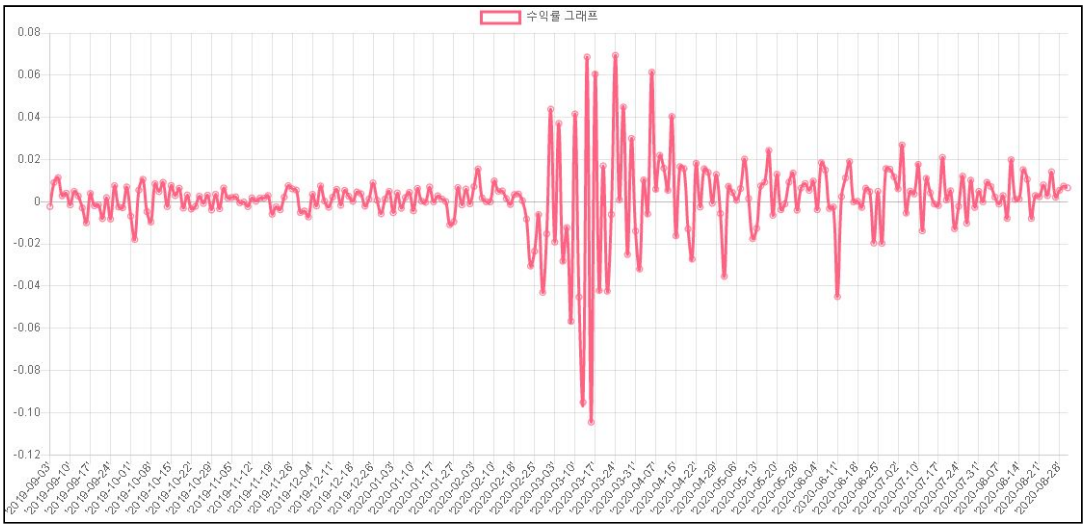
4.1 기본 M-V 모델

아래 이미지는 기본 모델의 main page 입니다. 사용자는 fama french의 5 industry에 기반한 5가지의 산업군(Cnsmr, Manuf, HiTec, Hlth, Other) 중 원하는 산업군을 선택할 수 있으며, 한 달, 세 달, 여섯 달 중 원하는 Rebalancing 주기를 선택할 수 있습니다. 또한 원하는 기간을 시작일과 종료일을 변경하여 지정할 수 있습니다. 기본 M-V 모델 페이지에는 3가지 기능(수익률 조회, Efficient Frontier, Rebalancing)을 확인할 수 있습니다.



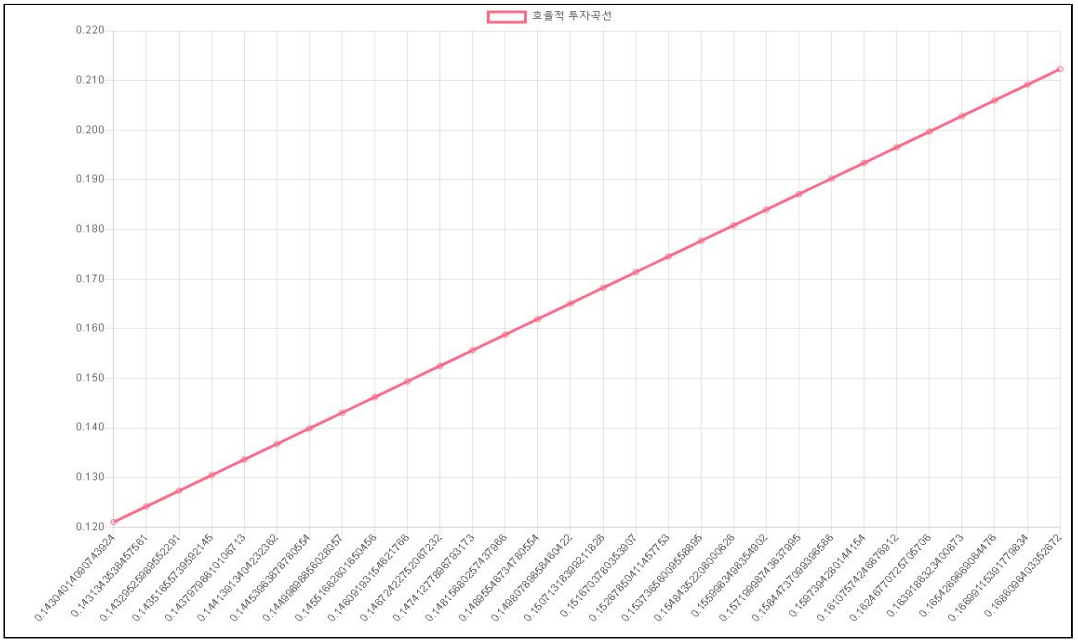
(1) 수익률 조회

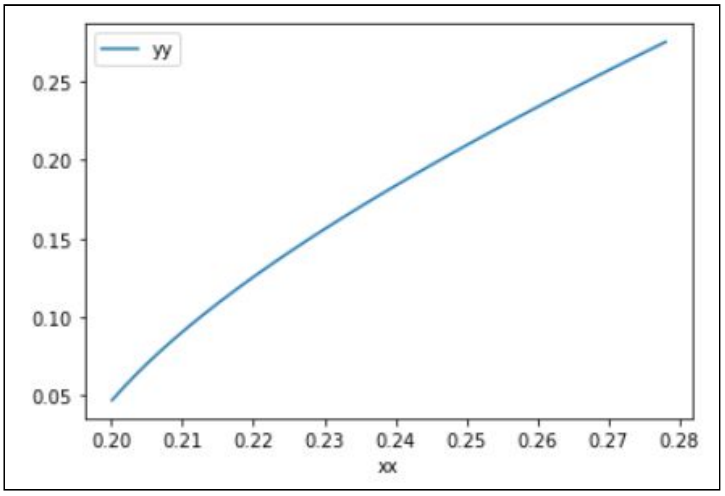
아래 이미지는 기본 M-V 모델 페이지에서 산업군은 Cnsmr를, 기간은 2019.09.01~2020.09.01을 선택하였을 때 수익률을 조회한 것입니다. x축은 일주일 단위의 기간을 의미하고, y 축은 수익률(adjusted Close)을 의미합니다.



(2) Efficient Frontier 조회

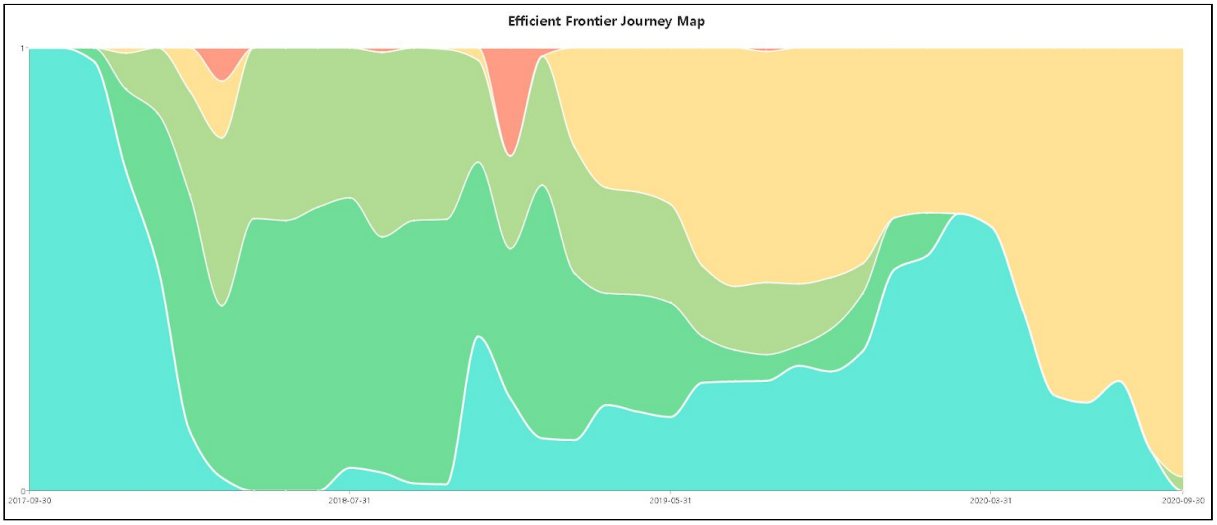
아래 이미지는 Efficient Frontier 페이지입니다. 5개의 산업군 모두를 사용하여 구현하였고, X축은 Risk를, y축은 Expected Return을 의미합니다. 아래 이미지에 나타난 그래프는 직선처럼 그려졌지만, 실제 같은 데이터를 가지고 파이썬에서 efficient frontier를 그렸을때는 아래 아래 이미지와 같이 곡선으로 나타남을 확인할 수 있습니다.





(3) Rebalancing 조회

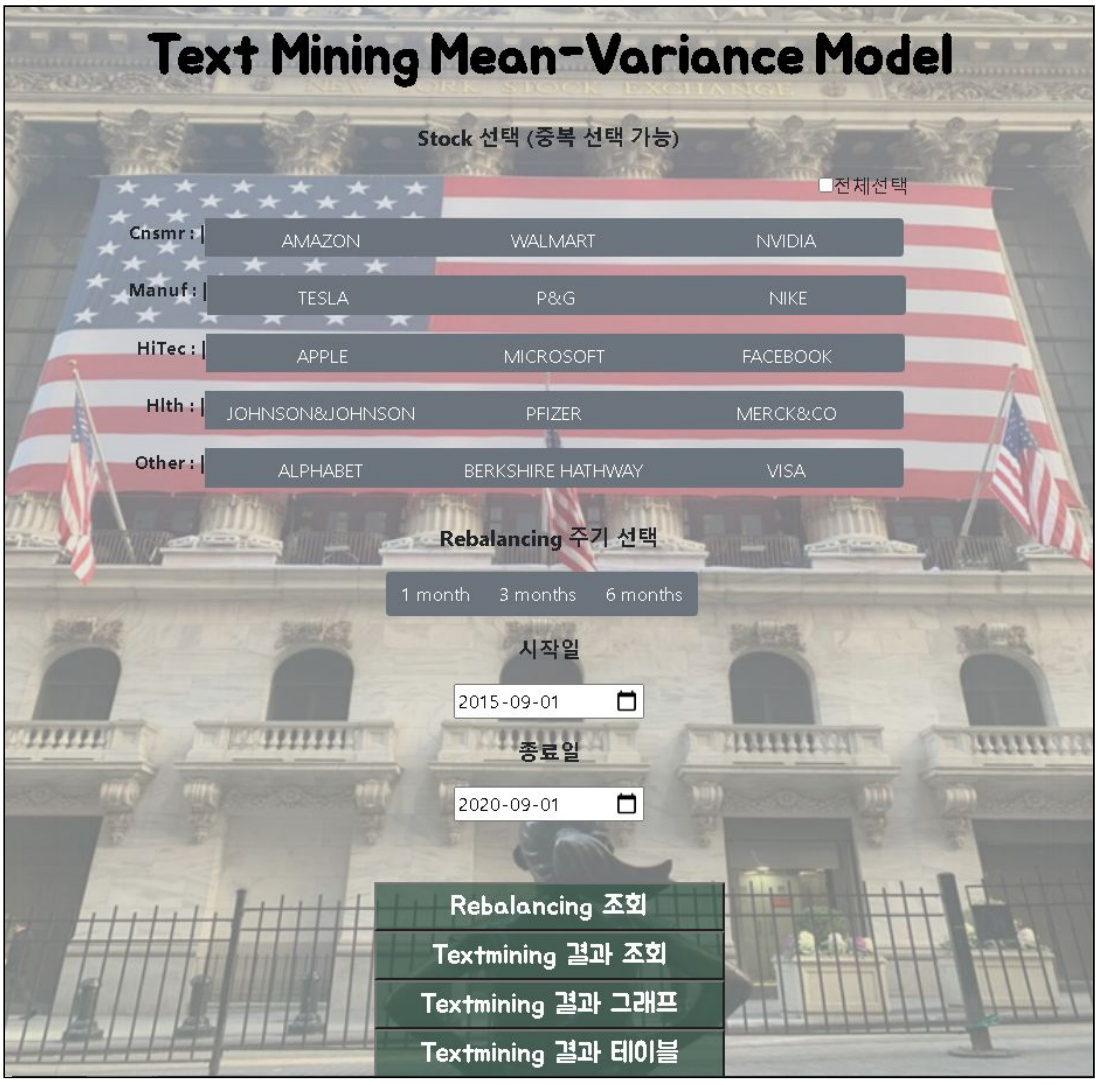
아래 이미지는 Rebalancing 페이지입니다. 5개의 산업군 모두를 사용하여 구현하였습니다. 기간은 2015.09-01~2020.09.01로 설정한 결과입니다.아래 이미지에서 각각의 색깔(5가지 색깔)은 각각 5가지의 산업군을 의미합니다. 해당 기능의 경우 텍스트 마이닝의 Rebalancing 기능과 겹치기 때문에 자세한 설명은 4.2 (1)에서 기술하겠습니다.



4.2 텍스트마이닝 Portfolio 시연

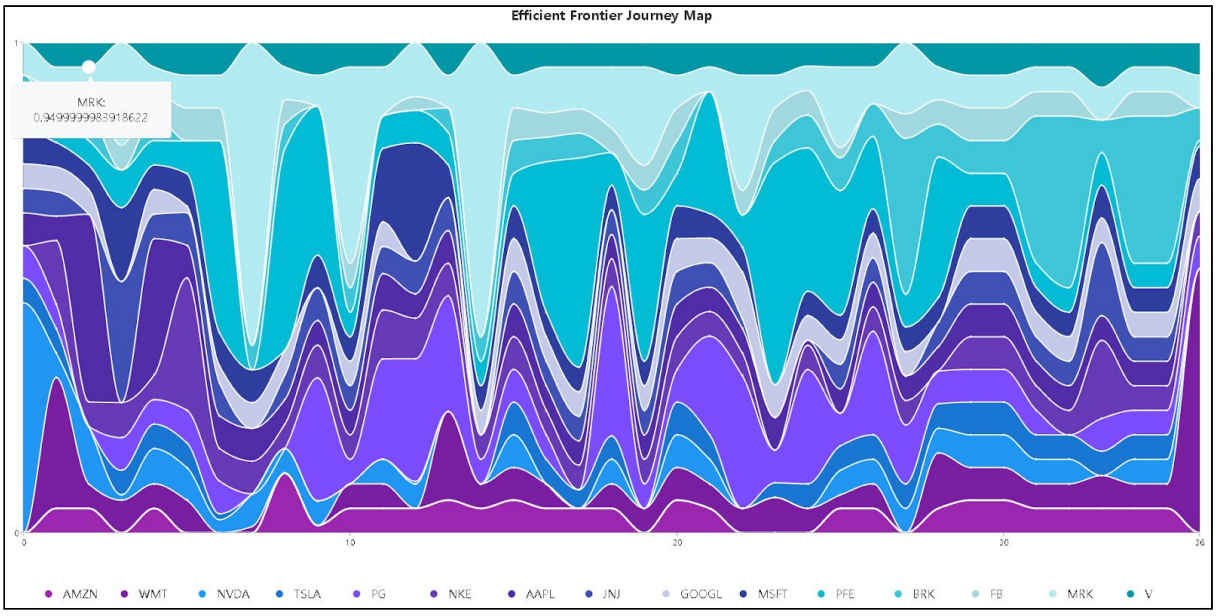
아래 이미지는 Text Mining 모델의 main page 입니다. 사용자는 15가지의 Stock(Amazon, Walmart, Nvidia, Tesla, P&G, Nike, Apple, MicroSoft, Facebook, Johnson & Johnson, Pfizer, Merck & Co,Alphabet, Berkshire hathaway, Visa) 중 원하는 Stock을 선택할 수 있으며, 한 달, 세 달, 여섯 달 중 원하는 Rebalancing 주기를 선택할 수 있습니다. 또한 원하는 기간을 시작일과 종료일을 변경하여 지정할 수 있습니다. 텍스트마이닝 M-V 모델 페이지에는 4가지 기능(Rebalancing조회, Text

Mining 결과 조회, Text Mining 결과 그래프, Text Mining 결과 테이블)을 확인할 수 있습니다.

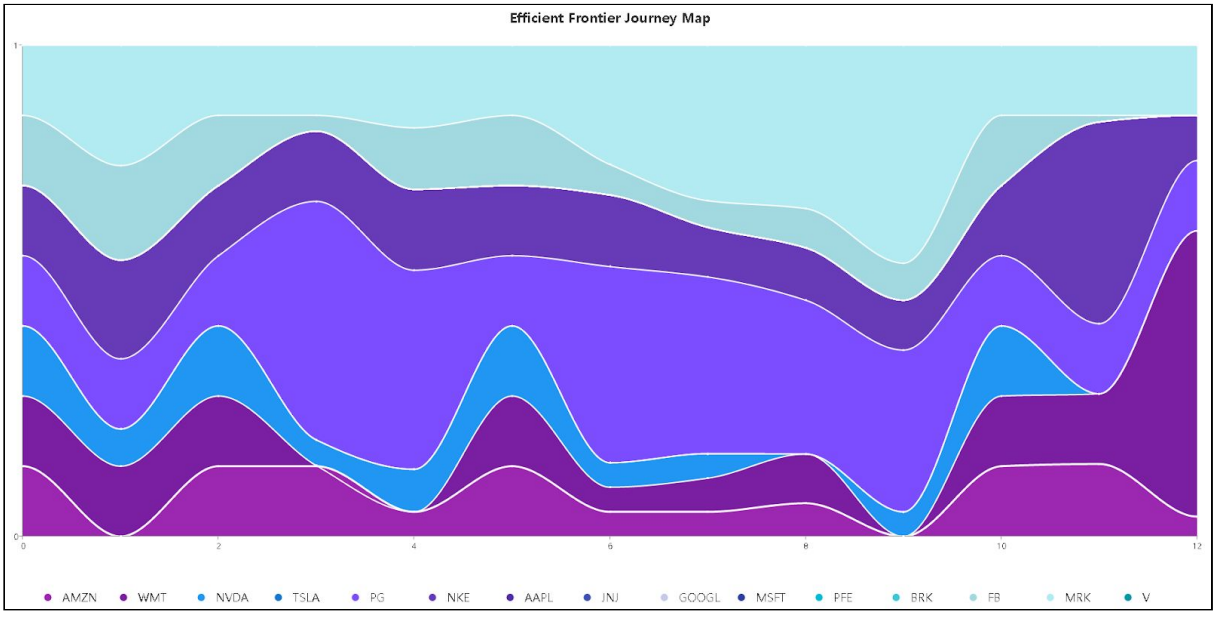


(1) Rebalancing 조회

아래 이미지는 Text Mining 모델의 Rebalancing 결과입니다. Rebalancing 은 설정한 기간 간격으로 Rolling을 하면서 최적화를 통해 선택된 stock들의 weight를 시각화한 것입니다. 이때 최적화는 GMV로 산출하였습니다. 사용자가 Stock의 종류(복수선택 가능), Rebalancing 주기, 전체 기간 등을 설정할 수 있기 때문에 두가지 경우로 예시를 준비하였습니다.
첫번째로 15개의 Stock을 모두 선택하고, Rebalancing 주기는 한달, 기간은 2015.09.01~2020.09.01로 설정하였습니다. Lookback 기간을 2년으로 설정하였기 때문에 아래 결과를 살펴보면 실제로는 2017.09월부터 2020.09월의 총 36개월의 데이터를 살펴볼 수 있습니다.



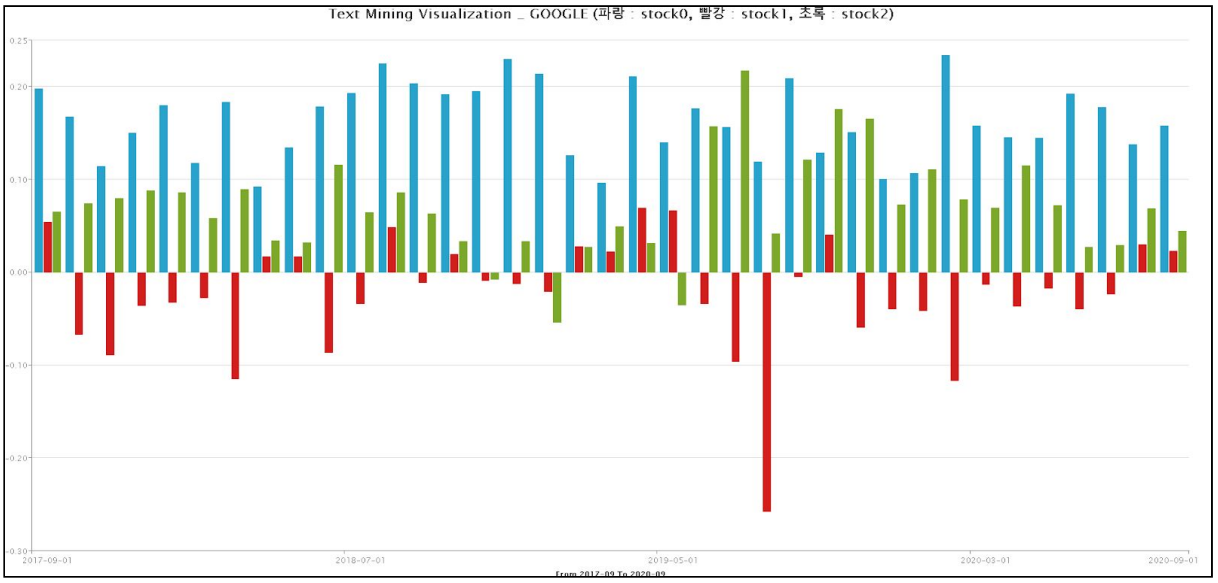
두번째 예시는 Stock을 7개(Amazon, Walmart, Nvidia, P&G, Nike, Facebook, Merck & Co) 선택하고, rebalancing을 3개월, 기간은 2016.09.01~2020.09.01로 선택한 경우입니다. 총 24개월 중 3개월마다 Rebalancing을 하기 때문에 8개 지점의 weight들을 살펴볼 수 있습니다.



(2) TextMining 결과 조회

아래 그래프는 텍스트마이닝의 mean 값을 기준으로 그린 그래프입니다. 해당 그래프는 Google News에서 각각 아마존 , 월마트, 엔비디아를 선택한 결과입니다. 아래 기간 동안 아마존은 부정적인 기사가 많았고, 월마트와 엔비디아는 긍정적인 기사가 많았음을 확인할 수 있습니다. Stock 별로 감성 분석의 편향이 존재함을

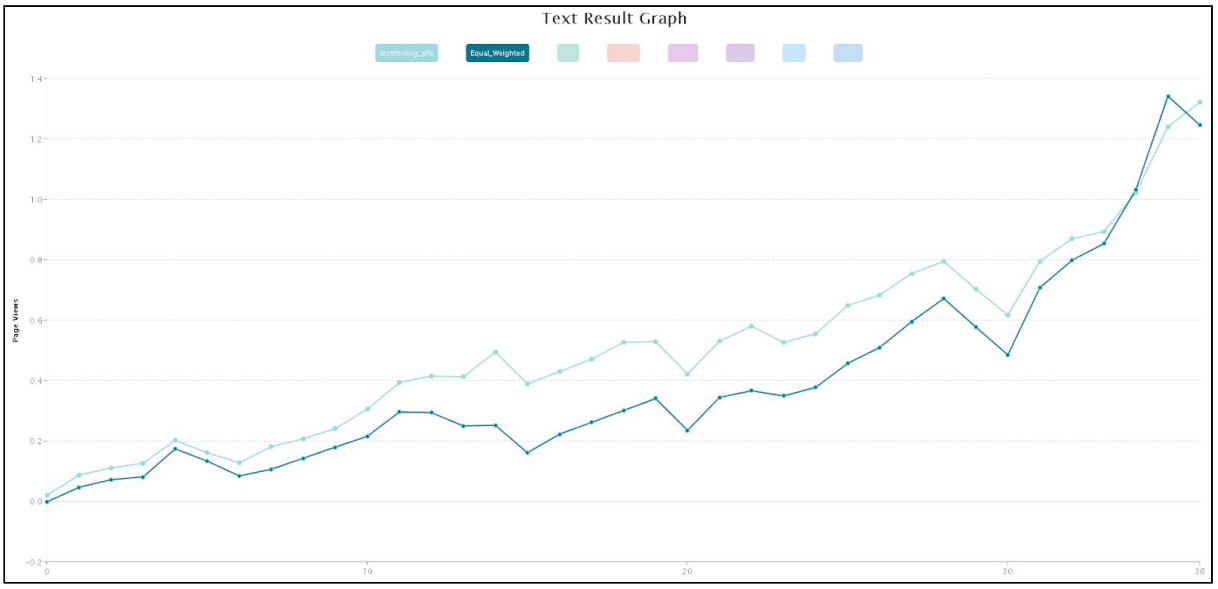
확인하였고 정규화를 진행하게 되었습니다.



(3) Text Mining 결과 그래프

아래 이미지는 Text Mining 결과 그래프를 나타낸 것입니다. 해당 그래프는 텍스트마이닝 포트폴리오와 7개 벤치마크의 누적 수익률입니다. SNP와 Equal-weight 산출에 사용된 수익률 데이터는 yahoo로 부터, 5 industry의 수익률 데이터는 fama french로부터 가져와 사용하였습니다.

아래 그래프는 텍스트마이닝 포트폴리오와 동일비중 포트폴리오만을 남겨놓은 그래프인데, 전반적 기간에 걸쳐 텍스트마이닝 포트폴리오 모델의 수익률이 동일비중 포트폴리오에 비해 우수함을 확인할 수 있었습니다.



(4) Text Mining 결과 테이블

아래 표는 4.1(3) 에서 사용하였던 텍스트마이닝 포트폴리오와 7개 벤치마크에 대하여 5가지의 평가지표를 표로 만든 것입니다. 5가지의 평가지표는 MDD, Sharpe Ratio, VaR, Mean, std를 제작하였습니다. 평가지표의 생성 방법에 대해서는 본 보고서의 4.2 평가지표 생성에서 자세히 기술하였습니다.

이때, Sharpe Ratio의 경우 시작 시점의 Risk free return을 사용하여 산출하였고, VaR의 경우 하위 5%의 결과를 사용하엿습니다. 또한 Mean과 std의 경우 각각 monthly 데이터에 12와 루트 12를 곱하여 산출하였습니다.

텍스트 마이닝 포트폴리오와 동일가중 포트폴리오의 결과가 가장 좋음을 확인할 수 있습니다.

index	textmining_pfo	Equal_Weighted	SNP	Cnsmr	Manuf	HiTec	Hlth	Other
MDD	0.09912821455119626	0.11174465088045 27	0.2000105039115498	0.16475528999999997	0.29833065596	0.16317802000000006	0.12259472893600004	0.2779146337999999
Sharpe ratio	1.860282089539033	1.4847454392445805	0.62213493562523	0.9407980676432239	0.13447729499957575	1.1199422459436652	0.6762545563233104	0.3361344445234657
VaR	-0.04741300172295721	-0.06411196286458887	-0.07236406432707006	-0.07224710027928008	-0.09161679693332402	-0.07234128691479633	-0.06474667928718171	-0.08785626419341185
Mean	0.28767602823647465	0.2819387539459459	0.11541539740540542	0.18103783783783786	0.03476756756756757	0.22232432432432434	0.1137081081081081	0.0745945945945946
Std	0.1524144173927409	0.18704714672017866	0.17503768531555505	0.18646369015909703	0.20179460657619974	0.1940105950482885	0.1584702370668891	0.20085172828919534

4.2 평가 지표 생성

파이썬 코드를 통해 장고에서 텍스트 마이닝 포트폴리오와 7가지 벤치마크에 대해 성능평가를 실시하였고, 이때 평가척도는 MDD, sharpe ratio, VaR, Mean, Std를 활용하였습니다. 각각의 평가척도는 각각의 Historical Return을 바탕으로 계산되었으며, VaR의 경우 95%의 케이스를 활용하였고, Mean은 Monthly Return에 12를 곱한 Arithmetic yearly return을 사용하였고 Std도 마찬가지로 sqrt(12)를 곱하여

산출하였습니다. sharpe ratio의 경우 텍스트 마이닝을 활용할 수 있는 가장 과거인 2017년도의 risk free rate를 기준으로 하였습니다. 앞서 설명한 평가 지표들은 각각 파이썬에서 def를 통해 생성되었으며, 위 사진과 같이 모든 지표를 한번에 보여주면 def를 통해 실제 장고의 프론트에서 출력하게 만들었습니다. 여기서 모든 def의 인풋데이터는 ret이며 수익률로 데이터를 인풋으로 받습니다.

```
performance = {'MDD': text_eval.mdd(input,ret),
               'Sharpe ratio': text_eval.sharpe_ratio(input,ret),
               'VaR': text_eval.value_at_risk(input,ret),
               'Mean': text_eval.Arithmetic_Mean_Annual(input,ret),
               'Std': ret.std() * np.sqrt(12)}
```

5. 한계점

분석 간 한계점이 여러가지 있었습니다.

첫째는 데이터의 양문제입니다. 텍스트 마이닝의 데이터 수집기간은 2017년 09월 부터 2020년 09월 까지 이루어졌는데 기간이 짧은 이유는 구글 뉴스를 크롤링 할 때, 원하는 기간을 설정하게 되면 path를 제공하지 않아 빠르게 크롤링을 진행하기 힘들었습니다. 따라서 사용자 관점 방식으로 실행시키는 검색엔진을 제작하여 구글 뉴스 크롤링을 진행하였습니다. 해당 방법의 경우 크롬 브라우저에서 크롤링 방지를 위해 로봇 자동화 검사 등을 실시하였기에 이를 회피하기 위한 알고리즘을 제작하였고 이로 인해 크롤링 시간이 굉장히 오래 소요되었습니다. 따라서 텍스트마이닝 수집기간을 3년으로 짧게 설정할 수 밖에 없었습니다.
짧은 데이터 길이로 인해 train set과 test set으로 나누어 평가하지 못했으며, 포트폴리오 투자기간 또한 2017년 09월 부터 2020년 09월 까지 3년의 기간만 고객이 설정할 수 있었습니다.

둘째, 데이터의 간격문제입니다. Seeking alpha 사이트의 경우 종목별 일별 기사수가 없거나 매우 적은 케이스가 종종있어서 월별 기사의 정보를 사용하였으며, 이로인해 월단위 보다 촘촘한 분석은 힘들었습니다. 추후 일별 데이터가 필요한 경우 Seeking alpha사이트 외에 다른 사이트 또한 고려해야 할 것 같습니다.

셋째, 종목의 감성이 주가에 주는 영향을 볼 때 시차를 고려하지 않았습니다.
감성분석의 경우 시차가 발생할 수 있는데 본 연구에서는 이를 고려하지 않았고, 추후 이러한 시차를 연구한 선행연구를 참고하여 추가 분석을 진행 할 수 있습니다. 이때, day단위로 시차를 연구한 선행연구와[9] min, hour 단위로 시차를 연구한 선행연구[10]가 있었습니다.

넷째, 텍스트 마이닝을 포트폴리오에 적용한 방법론에 대한 근거 부족. 연구 초기 찾은 선행논문[11]은 텍스트 마이닝 결과를 최적화 포트폴리오에 적용하긴 하였으나,

---

[9] Xie, Yancong, and Hongxun Jiang. "Stock market forecasting based on text mining technology: A support vector machine method." *arXiv preprint arXiv:1909.12789* (2019).
[10] Creamer, Germán G. "Can a corporate network and news sentiment improve portfolio optimization using the Black–Litterman model?." *Quantitative Finance* 15.8 (2015): 1405-1416.
[11] Sun, Andrew, Michael Lachanski, and Frank J. Fabozzi. "Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction." *International Review of Financial Analysis* 48 (2016): 272-281.

감성분석이 아닌 단순 언급빈도를 취합하여 이를 반영한 논문이었습니다. 앞서 언급한 시차관련 논문을 찾다 감성분석을 Black-Litterman 모델에 적용한 논문을 찾았고, 추후 이를 참고하여 추가분석을 진행해도 될것 같습니다. 적용방법을 간단히 설명하면, 첫째, 특정 index(STOXX 50)와 연관된 기업의 뉴스를 수집 둘째, negative, neutral, positive로 감성분석 및 최적의 시차 파악 셋째, long only portfolio에 negative, neutral, positive를 각각 0,1,2로 넣어 investors' view를 생성하였다.

6. REFERENCE

1. McCracken, Michael W., and Serena Ng. "FRED-MD: A monthly database for macroeconomic research." Journal of Business & Economic Statistics 34.4 (2016): 574-589.
2. McCracken, Michael W., and Serena Ng. "FRED-MD: A monthly database for macroeconomic research." Journal of Business & Economic Statistics 34.4 (2016): 574-589.
3. https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html
4. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
5. Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." arXiv preprint arXiv:1908.10063 (2019).
6. Smales, Lee A. "News sentiment in the gold futures market." Journal of Banking & Finance 49 (2014): 275-286.
7. Arvanitis, Konstantinos, and Nick Bassiliades. "Real-time investors' sentiment analysis from newspaper articles." Advances in combining intelligent methods. Springer, Cham, 2017. 1-23.
8. Sun, Andrew, Michael Lachanski, and Frank J. Fabozzi. "Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction." International Review of Financial Analysis 48 (2016): 272-281.
9. Xie, Yancong, and Hongxun Jiang. "Stock market forecasting based on text mining technology: A support vector machine method." arXiv preprint arXiv:1909.12789 (2019).
10. Creamer, Germán G. "Can a corporate network and news sentiment improve portfolio optimization using the Black–Litterman model?." Quantitative Finance 15.8 (2015): 1405-1416.
11. Sun, Andrew, Michael Lachanski, and Frank J. Fabozzi. "Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction." International Review of Financial Analysis 48 (2016): 272-281.

7. 추가 분석

지난번 발표때 산업군 내에서 흐름이 얼마나 비슷한지 물어보셨는데 이에 대해 hit map과 상관분석을 진행하였지만 특정한 패턴은 보이지 않았습니다. 우선 총 텍스트 마이닝 변수는 10개인데 이 중 포트폴리오에 사용된 변수는 p값이 가장 우수한 변수를 사용하여 종목마다 다른 변수가 적용되었고 이러한 점이 다른 이유일 수 있을 것 같습니다. 추후 같은 변수를

사용하거나 현재 월별 데이터 분석에서 일별 데이터를 적용 등을 통한 추가 분석이 필요한것 같습니다.